UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/436,618	11/09/1999	MICHAEL A. B. PARKES	200352	6893

7590 02/12/2003

LEYDIG VOIT & MAYER LTD
TWO PRUDENTIAL PLAZA SUITE 4900
180 NORTH STETSON
CHICAGO, IL 606016780

EXAMINER
ALI, SYED J

ART UNIT	PAPER NUMBER
2127	

DATE MAILED: 02/12/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/436,618	**Applicant(s)** PARKES ET AL.	
	Examiner Syed J Ali	**Art Unit** 2127	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)□ Responsive to communication(s) filed on _____ .

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-38 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-38 is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)□ The proposed drawing correction filed on _____ is: a)□ approved b)□ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)□ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____ .

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) □ The translation of the foreign language provisional application has been received.

15)□ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☒ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 2 .

4) □ Interview Summary (PTO-413) Paper No(s). _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: .

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1, 6-7, 10-11, 28, 32-33, and 36-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Doi et al. (USPN 6,298,382) (hereinafter Doi) in view of Nelson (USPN 5,452,447) (hereinafter Nelson).

As per claim 1, Doi teaches of a method of carrying out a procedure on a computer system having a memory, the memory containing user context data and global data, comprising:

executing a first server, wherein the first server defines a computer-executable function for performing a first sub-task of the procedure (col. 7 lines 44-53, "a plurality of processors share the object data to achieve a processing throughput", wherein Doi relates prior art which consists of dividing an application among many processing servers, with a separate engine to perform memory retrieval);

manipulating the global data to carry out the first sub-task (col. 6 lines 58-65, col. 7 lines 44-53, "a plurality of processors divisionally take charge of the object or candidate data to gain a response speed", wherein Doi relates prior art which shows how a client server may perform operations on global data and then ask for a retrieval server to retrieve data from memory in order to achieve a speedup);

sending the user context data to a second server (col. 6 lines 58-65, "master 10 serves as a retrieval managing server … to receive a retrieval request from a client", wherein the client submits a retrieval request to the retrieving server);

executing the second server, wherein the second server defines a computer-executable function for performing a second sub-task of the procedure (col. 7 lines 6-12, "slaves … take charge of the actual retrieval processing on databases 21A to 21D in accordance with an instruction [retrieval request] from the master", wherein upon the client server needing to retrieve data from memory, the search servers execute that request, wherein the retrieval process is interpreted to be a second sub-task of the procedure); and

manipulating the global data to carry out the second sub-task using the user context data (col. 7 lines 6-12, wherein the user data from the client server tells the search server what data to look for).

Doi does not specifically teach that the first and second servers are optimized to execute in the cache. However, Nelson discloses a caching file server, wherein a distributed processing system multiple clients are given access to a cache server (fig. 2 elements 12, 20). A caching server is provided so that memory accesses hit in the cache, and thus reduce the time required to access data. It would have been obvious to one of ordinary skill in the art to combine Doi and Nelson since the method of Doi allows for a second server to perform all memory accesses, but fails to explicitly state that the data retrieved should be placed in cache. To utilize the caching file of Nelson would easily translate in Doi in that the second server would perform the memory read, and then place the retrieved data into cache. In addition to the benefits of parallelism provided in Doi, a further improvement comes in using the caching file server of Nelson in the sense that if the same data is to be accessed subsequently, the data is already present in cache, thereby greatly reducing the time required for execution of the procedure.

As per claim 6, Doi and Nelson teach the method of claim 1, wherein the computer system has a first CPU and a second CPU, and the cache is comprised of a first area usable by the first CPU and a second area usable by the second CPU, and the executable code of the first server is optimized to fit in the first area and the executable code of the second server is optimized to fit in the second area.

Regarding the former limitations stating that the "computer system has a first CPU and a second CPU, and the cache is comprised of a first area usable by the first CPU and a second area usable by the second CPU", "Official Notice" is taken that cache is frequently located on the same integrated circuit as the CPU, and thus any processor would also contain a cache. Further, since Doi teaches of a method that utilizes more than one processing agent, thus at least two processors are combining to perform the functions described therein. Therefore, the method of Doi includes the limitations held herein.

Regarding the latter limitations stating that "the executable code of the first server is optimized to fit in the first area and the executable code of the second server is optimized to fit in the second area", Nelson teaches of a way of allowing an end user client to read and write data as well as query the attributes of a file through the use of a caching server (col. 7 line 57 – col. 8 line 14). Therefore, Nelson teaches of a way of allowing a computer system to execute in the cache. Further, Doi teaches a way of breaking a program into parallel execution by multiple servers. The combination of Nelson and Doi therefore permit a way of allowing a first processor to request retrieval from memory by a second server, and Nelson shows a way to optimize that execution by allowing it to be serviced in the cache, thereby improving execution time.

As per claim 7, Doi and Nelson teach the method of claim 1, wherein the procedure is a search of a database index tree containing a plurality of nodes, the first sub-task is to examine a node and the second sub-task is to perform an input/output operation for retrieving the node from memory and storing the node in the cache.

Doi relates an information retrieval method, which concerns a database retaining various kinds of information (col. 1 lines 8-15). Since the method of Doi is concerned more with the method of retrieving data and not with the format of the database, it would have been obvious to one of ordinary skill in the art that the method of Doi may relate to any number of database types, including the many that may be stored in a tree, such as heaps, binary trees, and splay trees, to name a few. Further, Doi discloses that a client server issues a request for information to be retrieved by a retrieval server (col. 8 lines 58-65). Doi does not specifically state that the node retrieved from memory should be stored in the cache.

However, Nelson teaches of a caching file server that allows read/write operations and queries to be serviced by the cache, as claimed (col. 7 line 57 – col. 8 line 14). It would have been obvious to one of ordinary skill in the art to combine Doi and Nelson since using the caching file server of Nelson would allow for faster data access, and the method of Doi permits a way of speeding up the retrieval of data from memory by allowing data access to be executed in parallel, while increasing the throughput of the application by allowing all memory access to be executed at once.

As per claim 10, Doi teaches the method of claim 1, wherein the computer system has a plurality of CPUs, and at least one server executes on only one CPU at a time (fig. 2 element 50, wherein the client could be any number of types of stations, but in this particular example is a CGI server. This is the client, which issues the search request, and executes on only one CPU. However, a plurality of CPUs is present, consisting of the client, master search manager, and a plurality of slave search servers.) The present invention concerns a first server, which issues a request to a second server. As it applies to the claim, the client computer 50 is the equivalent of the first server, and as disclosed in Doi, executes on only one CPU at a time.

As per claim 11, Doi teaches the method of claim 1, wherein the computer system has a plurality of CPUs, and at least two instances of one of the servers execute concurrently on different CPUs (fig. 2 elements 20A-20D, wherein a plurality of search servers work in combination to perform the search request issued by the client).

As per claim 28, regarding a computer-readable medium having computer-executable instructions for performing, on a computer system having a memory, user context data and global data, the method of claim 1, the methods described in Doi and Nelson are defined for a computer-readable medium having computer-executable instructions. Therefore, the discussion of claim 1 serves the basis for the rejection of this claim as well.

As per claim 32, regarding the computer-readable medium of claim 28, wherein the computer system has a first CPU and a second CPU, and the cache is comprised of a first area usable by the first CPU and a second are usable by the second CPU, and the executable code of the first server is optimized to fit in the first area and the executable code of the second server is optimized to fit in the second area, it is rejected for similar reasons as stated for claim 6.

As per claim 33, regarding the computer-readable medium of claim 28, wherein the procedure is a search of a database index tree containing a plurality of nodes, the first sub-task is to examine a node and the second sub-task is to perform an input/output operation for retrieving the node from memory and storing the node in the cache, it is rejected for similar reasons as stated for claim 7.

As per claim 36, regarding the computer-readable medium of claim 28, wherein the computer system has a plurality of CPUs, and at least one server executes on only one CPU at a time, it is rejected for similar reasons as stated for claim 10.

As per claim 37, regarding the computer-readable medium of claim 28, wherein the computer system has a plurality of CPUs, and at least two instances of one of the servers execute concurrently on different CPUs, it is rejected for similar reasons as stated for claim 11.

3. Claims 2-5, 8-9, 12-22, 29-31, 34-35, and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Doi in view of Nelson further in view of Ghodrat (USPN 6,425,021) (hereinafter Ghodrat).

As per claim 2, Ghodrat teaches of storing the user context data in a work packet and sending the work packet from the first server to the second server, wherein the work packet contains an action code for describing an action to be performed by the second server (col. 4 lines 36-50, "the asynchronous receive DMA is located within receive DMA (RDMA) unit 314 and contains two DMA contexts, a request handler and a response handler", wherein Ghodrat teaches of a way of directly accessing memory through the transferring of work packets of various contexts). Further, Doi teaches of clients sending requests for data retrieval and a second server performing that retrieval. However, Doi does not specifically mention how those requests are handled. It would have been obvious to one of ordinary skill in the art to use the system of Ghodrat for transferring packets since it is in the same field of endeavor, concerning memory access, and provides an efficient way of using a single interface to communicate several different contexts.

As per claim 3, Ghodrat teaches a system, wherein the work packet contains a reply state, and the method further comprises: causing the second server to update the work packet by replacing the value contained in the action code with the value contained in the reply state; and causing the second server to send the updated work packet back to the first server (col. 4 lines

36-50, “the asynchronous receive DMA is located within receive DMA (RDMA) unit 314 and contains two DMA contexts, a request handler and a response handler”, wherein Ghodrat provides a way of transferring packets, and provides a response handler for such a transaction). Further, as discussed above, Doi provides a way of handling requests and responses without specifying the mode of transferring data. The method of Ghodrat provides a way of defining a data structure that could be implemented in Doi to transfer retrieved data back to the client server upon issuance of the request for retrieval from memory.

As per claim 4, Ghodrat and Doi teach the method of claim 1, further comprising: in response to receiving a first work packet containing the user context data; causing the first server to partly perform the first sub-task using the first work packet; sending a second work packet containing the user context data from the first server to the second server; causing the second server to perform the second sub-task using the second work packet and store a result of the second sub-task in the second work packet; and sending the second work packet from the second server to the first server, wherein the result is useable by the first server to complete the performance of the first sub-task (col. 4 lines 36-50, “the asynchronous receive DMA is located within receive DMA (RDMA) unit 314 and contains two DMA contexts, a request handler and a response handler”, wherein Ghodrat describes a way of transferring data packets of different contexts through a single interface). Further, Doi teaches of a way of allowing a second server to perform memory access for a client, and communicating the results back and forth, although Doi does not specify the mode of that communication. It would have been obvious to use Ghodrat’s method of data transfer. The motivation for the combination of these references can be found above, as in the discussion of claims 2 and 3.

As per claim 5, Doi teaches the method of claim 4, wherein the second work packet is linked as a child to the first work packet (fig. 2 elements 10, 50, and 20A-20D, wherein a relationship is created among a client, master retrieval server, and slave retrieval servers). The relationship among these servers constitutes a way in which the slave retrieval servers respond to requests of the client to retrieve data from memory. The client/master/slave relationship can be interpreted also as a parent/child relationship, and further, to relate any objects in such a way is well known in the art. It would have been obvious to include some form of this relationship in the data packet, as described in Ghodrat for the purpose of making clear the directionality of the communication as well as defining the requests and responses.

As per claim 8, Doi in view of Nelson further in view of Ghodrat teaches the method of claim 7, further comprising:

in response to receiving a first work packet containing the user context data; causing the first server to determine if a node is in the cache ("Official Notice" is taken that when performing an operation on data stored in memory, it is well known and expected in the art that the processor would first look to see if that data were present in a block of cache); and

if the node is determined not to be in the cache, sending a second work packet containing the user context data from the first server to the second server. (Doi, col. 6 lines 58-65, wherein when data needs to be retrieved from memory, the client sends a request to the master search manager) (Ghodrat, col. 4 lines 36-50, "the asynchronous receive DMA is located within receive DMA (RDMA) unit 314 and contains two DMA contexts, a request handler and a response handler", wherein in addition to handling requests and responses which clearly apply to the method of Doi, Ghodrat also teaches of a method of transmitting these requests and responses via data packets);

causing the second server to retrieve the node from a main memory using the second work packet and store the node in cache (Doi, col. 7 lines 6-12, wherein upon issuance of a search request by the client, the search servers retrieve the data from memory, and as discussed for claim 7, Nelson teaches of a method for optimizing performance by allowing this process to be serviced in cache);

causing the second server to store a reference to the cached node in the second work packet (Doi, col. 7 lines 44-53, "a server returns the processing result to the client", wherein upon retrieval of the data from memory, the search server tells the client how to find the data, and in combination with the caching file server of Nelson, it would have been obvious to reference the location in cache at the time of returning the processing result); and

sending the second work packet from the second server to the first server, wherein the first server responds to the receipt of the second work packet by searching the cached node (Doi, col. 7 lines 44-53, "a server returns the processing result to the client", wherein the client server, after learning of the location of the desired data would look in the caching file server of Nelson and perform the necessary operations).

As per claim 9, see the discussion for claim 5 regarding the method of claim 8, wherein the first work packet contains a reference to a parent work packet. The parent/child relationship is described as it relates to Doi, and if the second work packet is linked as a child to the first work packet, it follows that the second work packet would contain some sort of reference to the parent. By this means, after the search servers perform the processing request, it is known what client made the request. If no reference to the parent work packet were contained, then there would be no way of sending a response to the requesting node.

As per claim 12, Doi teaches the method of claim 1, wherein the computer system has a first CPU and a second CPU, and the work packet has a designated value, and wherein one of the servers executes on the first CPU when the designated value falls within a first range and executes on the second CPU when the designated value falls within a second range. Specifically, Doi discloses prior art (fig. 9, col. 2 lines 55-62, "there frequently occurs a case that the retrieving server (processor) the retrieval data belongs to differs from the retrieving server (processor) which conducts the retrieval processing of that retrieval data", wherein when a retrieval request is submitted, there is communication between the retrieval servers so that when a request is issued for which the data falls in a particular range, the correct retrieval server performs the request). In this sense, this communication allows for specific processors to conduct retrieval on a specific range, and if that range is not allocated to that server, the communication link allows the request to be handled correctly.

As per claim 13, Doi in view of Nelson further in view of Ghodrat teach of a method of writing a computer program for carrying out a procedure on a computer system having a cache, global data and a user context comprising:

dividing the procedure into sub-tasks (Doi, col. 6 line 58 – col. 7 line 53, wherein a client server performs some portion of the work on an application, before sending search requests to a search manager to perform all memory retrievals, thereby dividing the procedure into at least two sub-tasks);

defining a server for each sub-task, wherein each server contains instructions for performing its respective sub-task on the global data and wherein each server is optimized to fit inside the cache when executed (Doi, col. 6 line 58 – col. 7 line 53, wherein the first server is a client which requests retrieval from memory and the second server is a master/slave search engine which performs the memory retrieval) (Nelson, col. 7 line 57 – col. 8 line 13, wherein Nelson describes how various operations on global data ["remote file" in the disclosure of Nelson] can be optimized to be serviced in the caching file server); and

defining a work packet for transferring the user context between two or more of the servers (Ghodrat, col. 6 line 25-48, "Link 324 sends packets, which appear at the various transmit FIFO interfaces", wherein Ghodrat describes a way of transferring data packets to convey work that needs to be done).

The motivation for combining Doi with Nelson and Ghodrat is discussed above in reference to claims 1-12.

As per claim 14, Doi discloses the method of claim 13, further comprising:

defining an action code to be located in the work packet for describing an action to be performed by a server (col. 6 lines 58-65, wherein Doi discloses that a requesting client submits a request to he master search manager. This request tells the master search manager what data to look up in the database. The "action code" could mean any of a number of ways of telling the master search manager what data to look for, and any number of implementations could serve this purpose).

As per claim 15, Doi teaches the method of claim 13, further comprising defining a reply state code for the work packet, the reply state code being useable by a server to gain information about the results of a function executed by another server (col. 7 lines 44-53, "a server returns the processing result to the client", wherein the search server submits a reply to the requesting client, telling that client what the results of the submitted search were).

As per claim 16, Doi and Ghodrat teach the method of claim 13, further comprising:

defining a first work packet for a first server (Ghodrat, fig. 3, and col. lines 25-56, wherein the Link 324 sends data packets across the network for servicing by another node on the network. The originating computer performs some amount of work before communicating its needs to another node in the network. This can be considered the first work packet. Additionally, in Doi, a client server performs some amount of work on a process before asking for help from a search manager. The original work done can be considered the first work packet);

defining a second work packet for a second server, wherein the first work packet is usable by the first server to perform a first sub-task on the global data and the second work packet is usable by the second server to receive the user context from the first server, perform the second task, and return a result of the second task to the first server. (Ghodrat, fig. 3, and col. lines 25-56, wherein the Link 324 sends data packets across the network for servicing by another node on the network. This packet being submitted can be considered the equivalent of the second work packet, and in combination with the parallel processing of Doi constitutes a way of communicating cooperative work to be done over more than one server via transmission of data packets).

As per claim 17, regarding the method of claim 16, wherein the second work packet is linked as a child to the first work packet, it is rejected for similar reasons as stated for claim 5.

As per claim 18, regarding the method of claim 13, wherein the computer system has a first CPU and a second CPU, and the cache is comprised of a first area usable by the first CPU and a second are usable by the second CPU, and the first server is optimized to fit in the first area and the second server is optimized to fit in the second area, it is rejected for similar reasons as stated for claim 6.

As per claim 19, regarding the method of claim 13, wherein the procedure is a search of a database index tree containing a plurality of nodes, the first sub-task is to examine a node and the second sub-task is to perform an input/output operation for retrieving the node from memory, it is rejected for similar reasons as stated for claim 7.

As per claim 20, regarding the method of claim 13, wherein the computer system has a plurality of CPUs, and at least one of the servers is defined to execute on only one CPU at a time, it is rejected for similar reasons as stated for claim 10.

As per claim 21, regarding the method of claim 13, wherein the computer system has a plurality of CPUs and at least one of the servers is defined to run concurrently as at least two instances on different CPUs, it is rejected for similar reasons as stated for claim 11.

As per claim 22, regarding the method of claim 13, wherein the computer system has a first CPU and a second CPU, a designated value field is defined for the work packet, and wherein at least one of the servers executes on the first CPU when the designated value falls within a first range and executes on the second CPU when the designated value falls within a second range, it is rejected for similar reasons as stated for claim 12.

As per claim 29, regarding the computer-readable medium of claim 28, having further computer-executable instructions for: storing the user context data in a work packet and sending the work packet from the first server to the second server, wherein the work packet contains an action code for describing an action to be performed by the second server, it is rejected for similar reasons as stated for claim 2.

As per claim 30, regarding the computer-readable medium of claim 29, wherein the work packet contains a reply state, and the computer-readable medium has further computer-executable instructions for: causing the second server to update the work packet by replacing the value contained in the action code with the value contained in the reply state; and causing the second server to send the updated work packet back to the first server, it is rejected for similar reasons as stated for claim 3.

As per claim 31, regarding the computer-readable medium of claim 28, having further computer-executable instructions for: in response to receiving a first work packet containing the user context data; causing the first. server to partially complete the first sub-task using the first work packet; sending a second work packet containing the user context data from the first server to the second server; causing the second server to perform the second sub-task using the second work packet and store a result of the second sub-task in the second work packet; sending the second work packet to the first server, wherein the result is useable by the first server to fully complete of the first sub-task, it is rejected for similar reasons as stated for claim 4.

As per claim 34, regarding the computer-readable medium of claim 33 having further computer-executable instructions for: in response to receiving a first work packet containing the user context data; causing the first server to determine if a node is in the cache; and if the node is determined not to be in the cache, sending a second work packet containing the user context data from the first server to the second server; causing the second server to retrieve the node from a main memory using the second work packet and store the node in cache; causing the second server to store a reference to the cached node in the second work packet; and sending the second work packet from the second server to the first server, wherein the first server searches the cached node, it is rejected for similar reasons as stated for claim 8.

As per claim 35, regarding the computer-readable medium of claim 34, wherein the first work packet contains a reference to a parent work packet, it is rejected for similar reasons as stated for claim 9.

As per claim 38, regarding the computer-readable medium of claim 28, wherein the computer system has a first CPU and a second CPU, and the work packet has a designated value, and wherein one of the servers executes on the first CPU when the designated value falls within a first range and executes on the second CPU when the designated value falls within a second range, it is rejected for similar reasons as stated for claim 12.

4. Claims 23-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Doi in view of Ghodrat.

As per claim 23, Doi and Ghodrat teach a computer-readable medium having stored thereon a data structure, the data structure comprising: a work packet for transferring user context information between at least two servers, wherein each server defines at least one function for performing a sub-task of a computer-executable procedure to manipulate a global data set (Doi, col. 6 line 58 – col. 7 line 53, wherein a procedure is divided in its execution among a client that performs a sub-task of the procedure, before sending a request to a master search manager to perform a second sub-task of the procedure, specifically retrieving requested data from memory) (Ghodrat, col. 6 lines 25-47, wherein a computer attached to a network submits requests via data packets across a network, and the remote device then processes the request and sends a response to the requesting node). Therefore, the combination of Doi and Ghodrat teach all the limitations of the claim. The motivation for using the data transmission medium of Ghodrat with the parallel processing method of Doi can be found above as it applies to claims 1-12.

As per claim 24, regarding the computer-readable medium of claim 23, wherein the work packet has defined therein an action code for describing an action to be performed by one of the servers, it is rejected for similar reasons as stated for claim 2.

As per claim 25, regarding the computer-readable medium of claim 24, wherein the work packet has defined therein a reply state usable by one of the servers to send a result of its sub-task to another server, it is rejected for similar reasons as stated for claim 3.

As per claim 26, Doi and Ghodrat teach of a computer-readable medium having stored thereon a data structure, the data structure comprising: a first server defining at least one function for performing a sub-task of a computer-executable procedure to manipulate a global data set, wherein the first server executes the function in response to the receipt of a first work packet, the work packet containing user context information usable by the server to perform the sub-task, and wherein the first server transmits the user context information to a second server using a second work packet (Doi, col. 6 line 58 – col. 7 line 53, wherein a procedure is divided in its execution among a client that performs a sub-task of the procedure, before sending a request to a master search manager to perform a second sub-task of the procedure, specifically retrieving requested data from memory) (Ghodrat, col. 6 lines 25-47, wherein a computer attached to a network submits requests via data packets across a network, and the remote device then processes the request and sends a response to the requesting node). Doi does not specifically mention that the first server executes the function in response to the receipt of a first work packet. However, Doi does not specifically mention the data structure used to contain the instructions to be executed. Ghodrat teaches of a transmission method for data packets to perform the work of the function. It is easily applicable to use this data structure in the method of Doi. Therefore, the combination of Doi and Ghodrat teach all the limitations of the claim. The motivation for using the data transmission medium of Ghodrat with the parallel processing method of Doi can be found above as it applies to claims 1-12.

As per claim 27, regarding the computer-readable medium of claim 26, wherein the second server stores a result of a second sub-task performed on the global data in the second work packet and returns the second work packet to the first server, it is rejected for similar reasons as stated for claim 8.

Conclusion

5. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (703) 305-8106. The examiner can normally be reached on Mon-Fri 8-5:30, 1st Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John A Follansbee can be reached on (703) 305-8498. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7239 for regular communications and (703) 746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Syed Ali
February 3, 2003

MAJID BANANKHAH
PRIMARY EXAMINER